Monday, July 18, 2022 9:01 PM

n length array [0 to n-1]

6

0, 1, 2, 3, 4, 5

T – O(n)

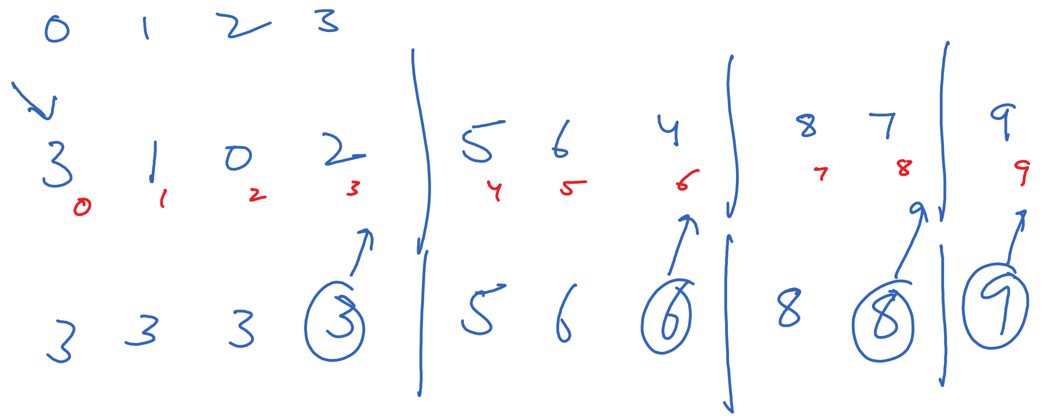
S – O(1)

(1, 2, 0) | (4, 3) | (5)

0 – 1 – 2 | 3 – 4 | 5

$\boxed{0-9}$

O(n) O(1)

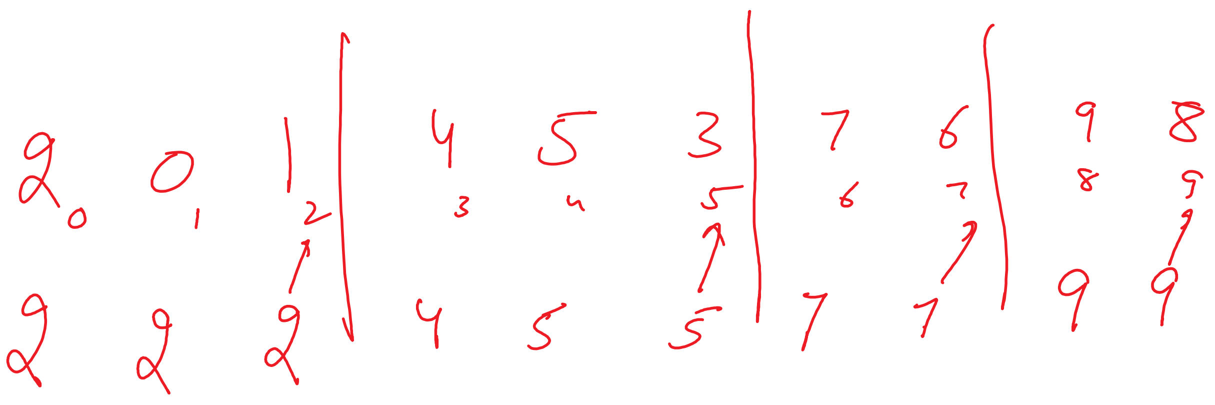


W?

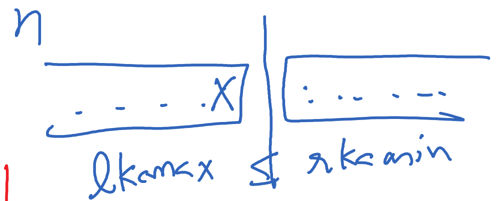
H?

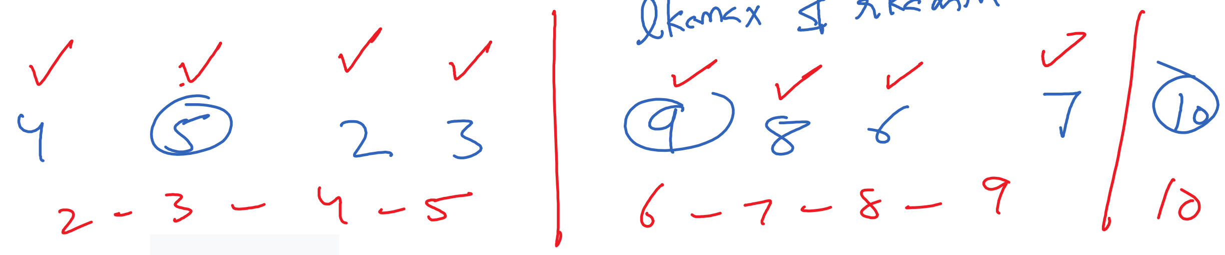
Why?

7 [0 to 6]

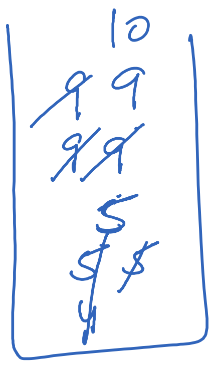


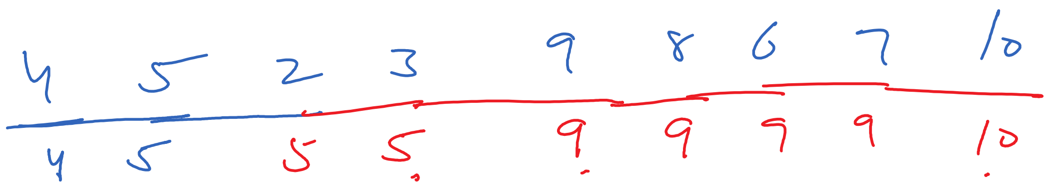
Max Chunks 2

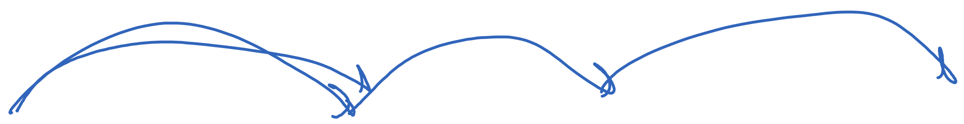




[2,1,3,4,4]







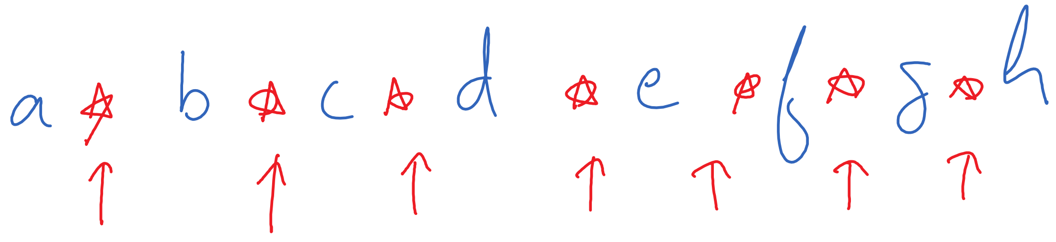
Max Chunk 2

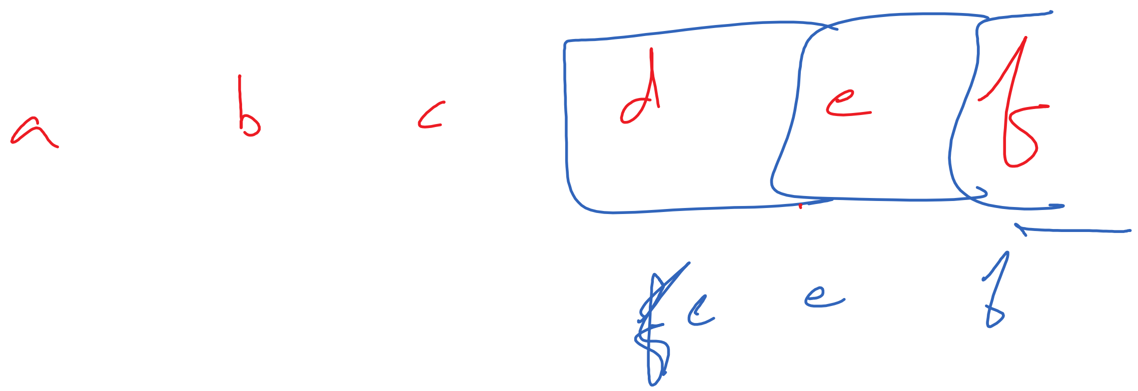
Time O(n) Space O(n)

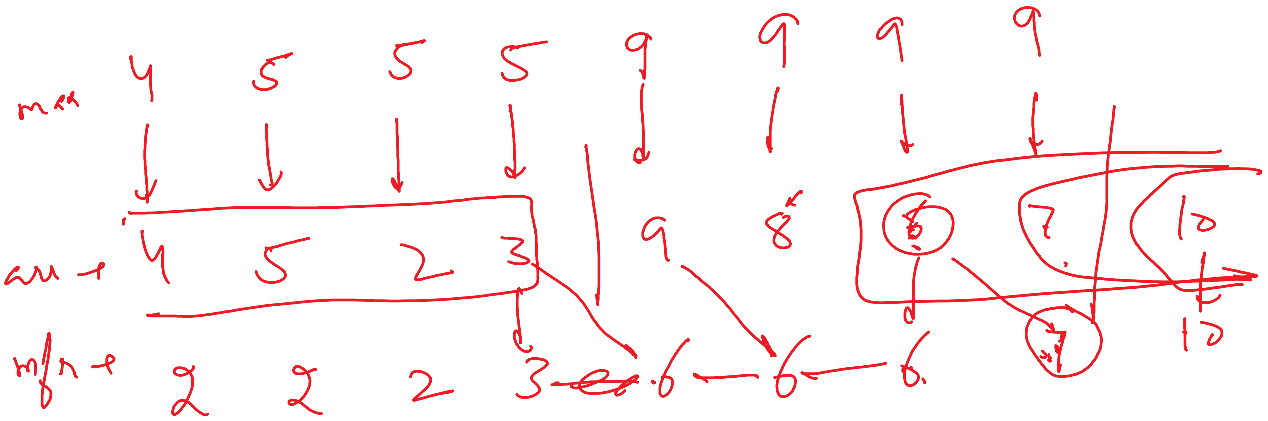
6 2 | 7 | 10 9 | 11

2 - 6 | 7 | 9 - 10 - 11

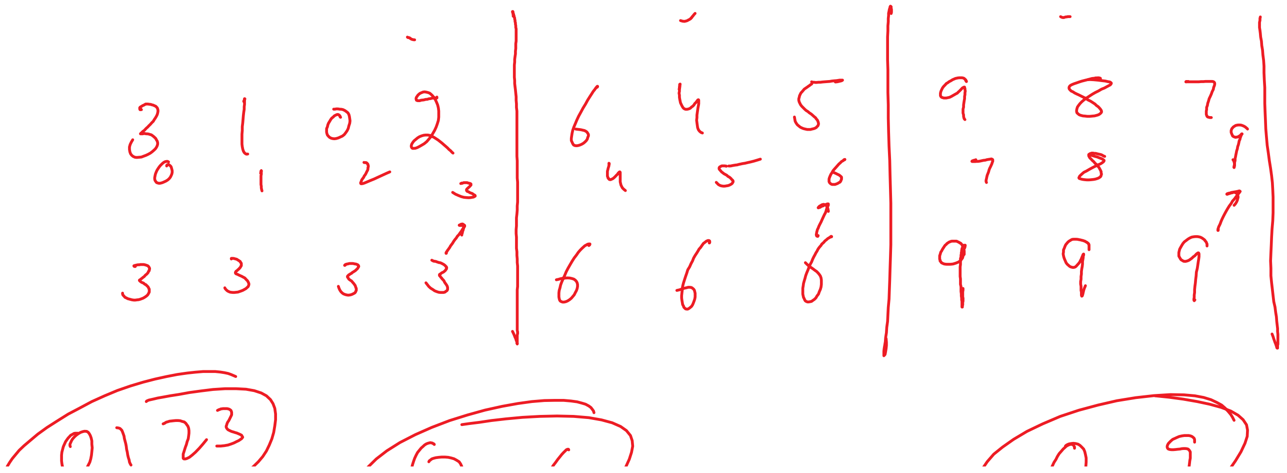
9:35 - 9:45




a
b
c
d
e
f
e
e
f


max
4 5 5 5 9 9 9 9
arr
4 5 2 3 9 8 6 7 10
min
2 2 2 3 6 6 6 7 10

10 [0 - 9]


3 1 0 2
0 1 2 3
3 3 3 3
6 4 5
4 5 6
6 6 6
9 8 7
7 8 9
9 9 9
0 1 2 3

0123 0-6 0-9

[2, 8]

```
[2,9,2,5,6], left = 2, right = 8
```

a b c d e f g h

2 9 2 5 6

[2] [2], [2,5], [2,5,6]

[5], [5,6]

[6]

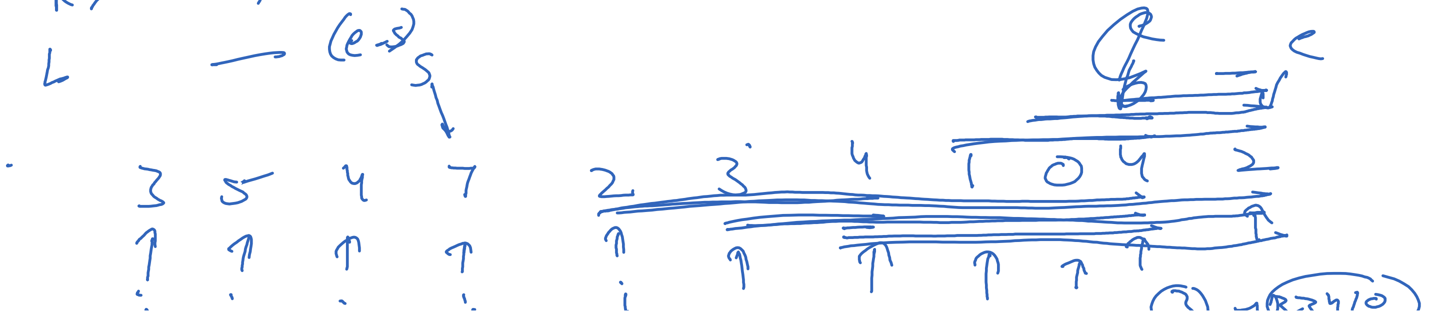
10:22 to 10:35

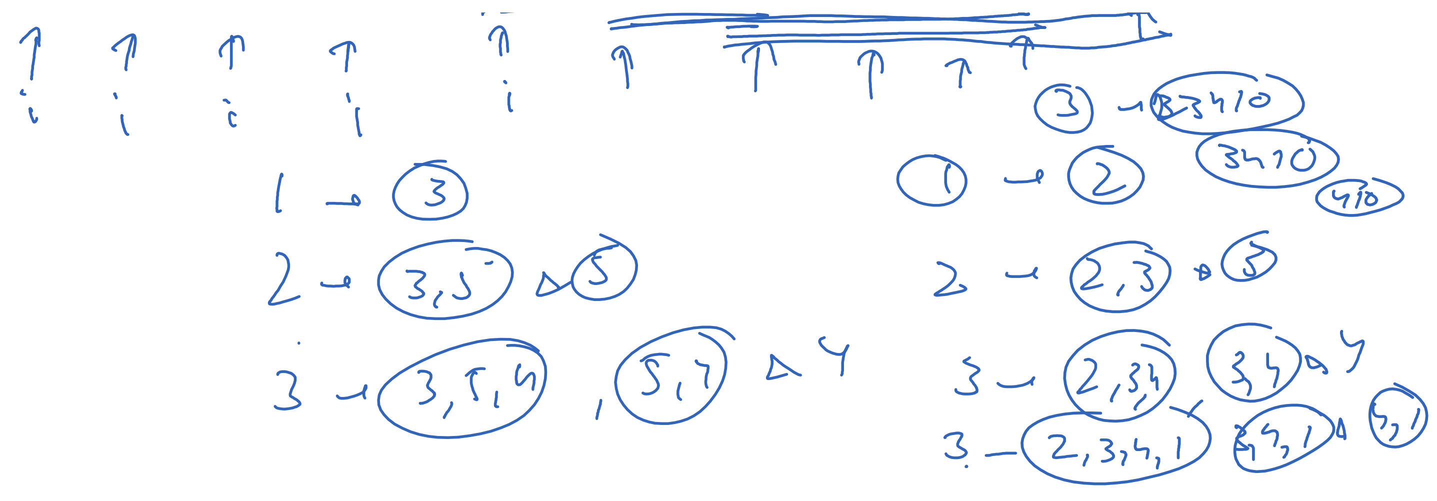
L, R e++ (e-s)

R> e, s=i (e-s)

L — (e-s)

2 - 5

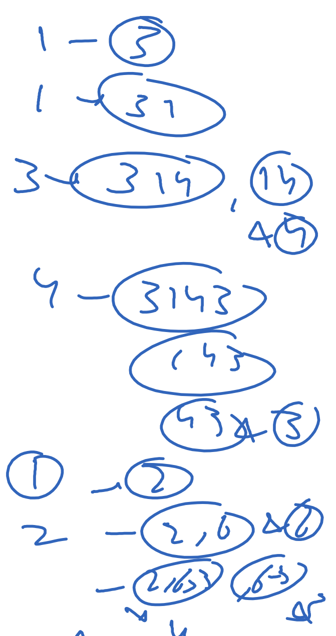


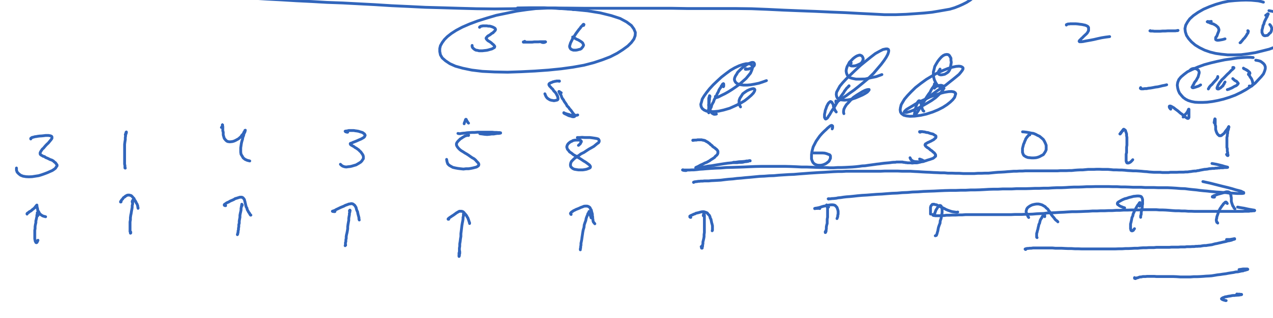


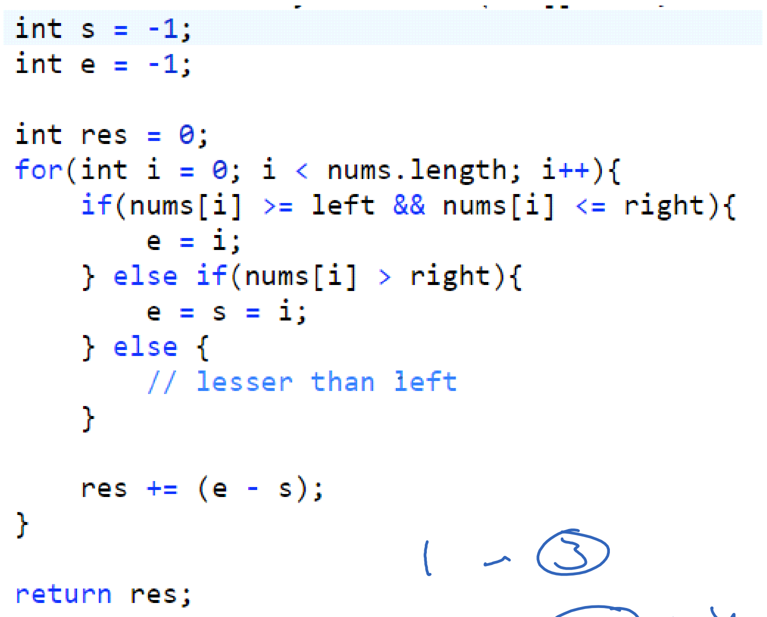
[L, R] e = i , $\boxed{e - s}$

> R e = s = i , $\boxed{e - s}$

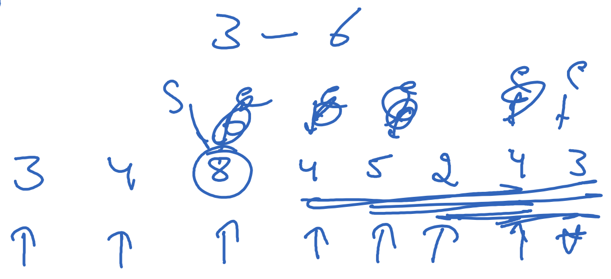
< L — , $\boxed{e - s}$

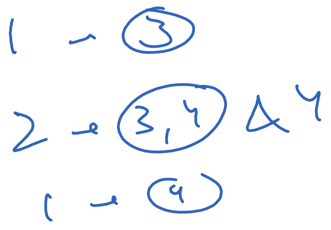


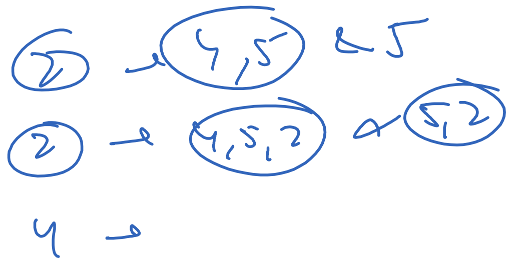


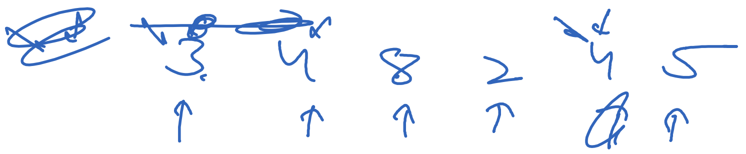


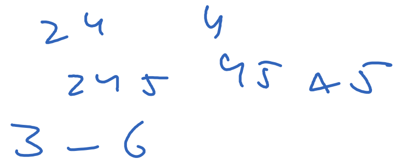
```
int s = -1;
int e = -1;

int res = 0;
for(int i = 0; i < nums.length; i++){
    if(nums[i] >= left && nums[i] <= right){
        e = i;
    } else if(nums[i] > right){
        e = s = i;
    } else {
        // lesser than left
    }

    res += (e - s);
}

return res;
```

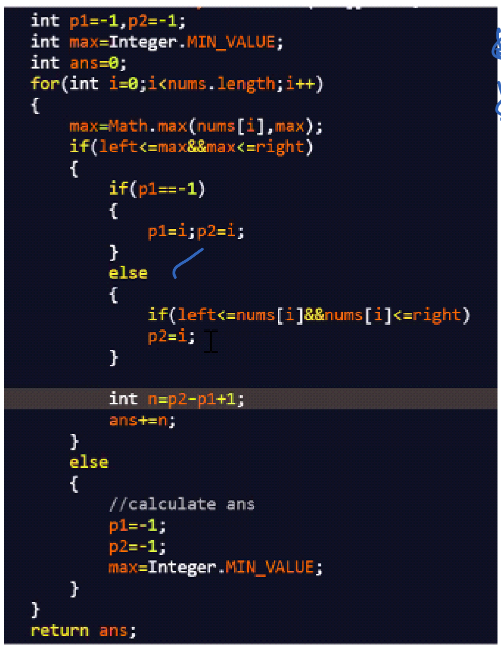




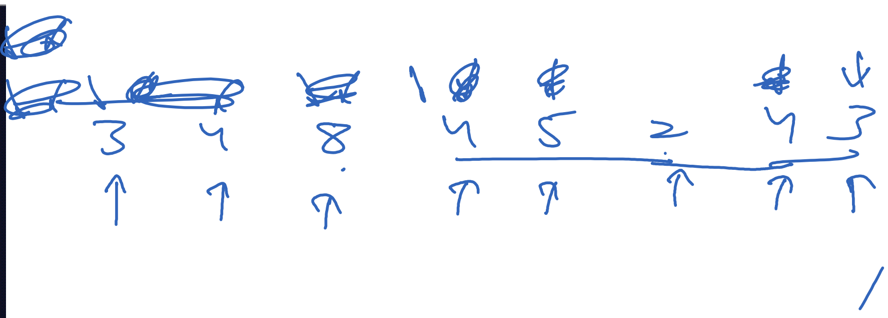


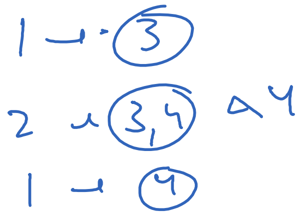


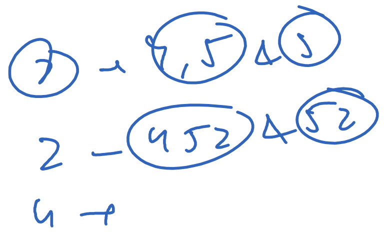


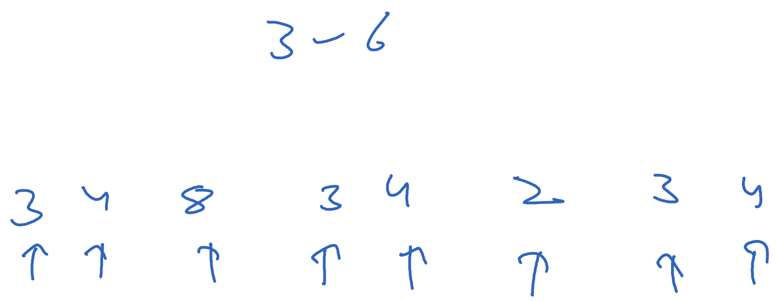


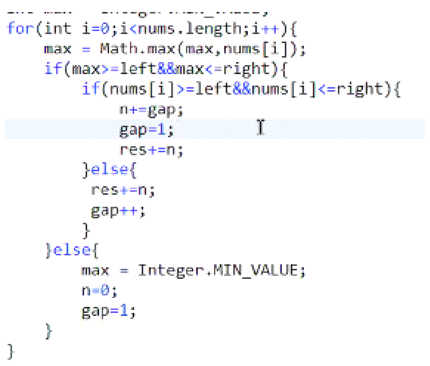
```
int p1=-1,p2=-1;
int max=Integer.MIN_VALUE;
int ans=0;
for(int i=0;i<nums.length;i++)
{
    max=Math.max(nums[i],max);
    if(left<=max&&max<=right)
    {
        if(p1==-1)
        {
            p1=i;p2=i;
        }
        else
        {
            if(left<=nums[i]&&nums[i]<=right)
            p2=i;
        }

        int n=p2-p1+1;
        ans+=n;
    }
    else
    {
        //calculate ans
        p1=-1;
        p2=-1;
        max=Integer.MIN_VALUE;
    }
}
return ans;
```



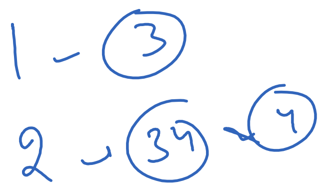


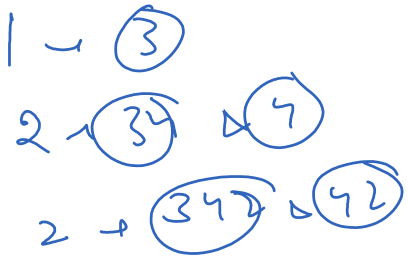


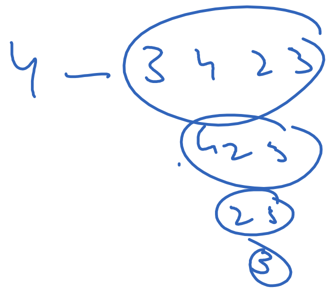


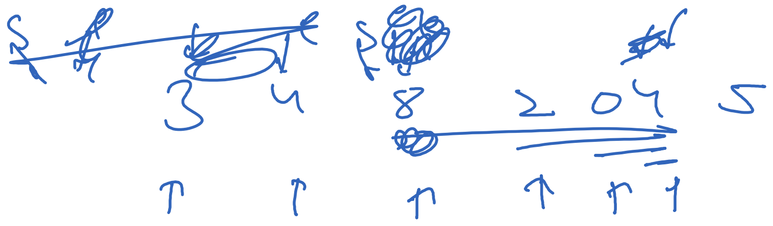


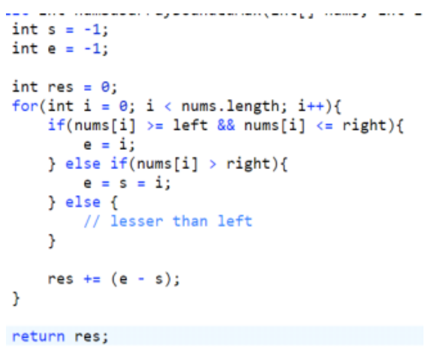
```
for(int i=0;i<nums.length;i++){
    max = Math.max(max,nums[i]);
    if(max>=left&&max<=right){
        if(nums[i]>=left&&nums[i]<=right){
            n+=gap;
            gap=1;
            res+=n;
        }else{
         res+=n;
         gap++;
        }
    }else{
        max = Integer.MIN_VALUE;
        n=0;
        gap=1;
    }
}
```



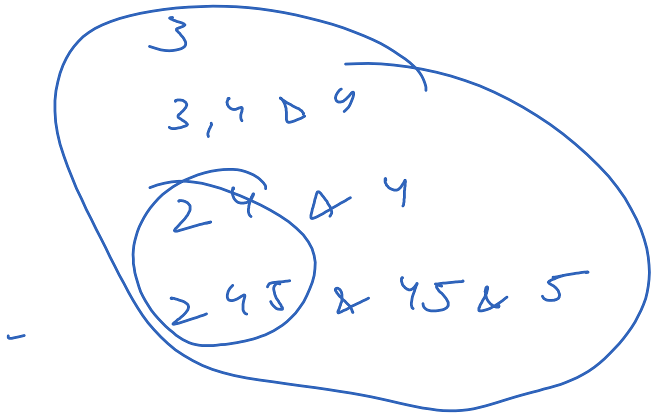


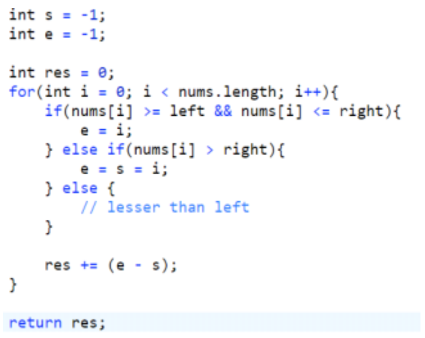




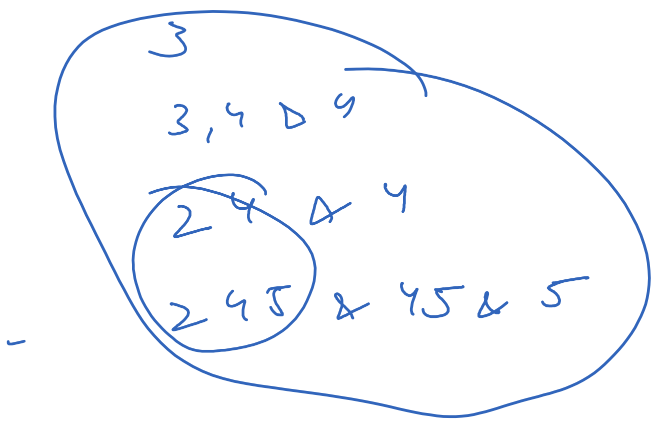


```
int s = -1;
int e = -1;

int res = 0;
for(int i = 0; i < nums.length; i++){
    if(nums[i] >= left && nums[i] <= right){
        e = i;
    } else if(nums[i] > right){
        e = s = i;
    } else {
        // lesser than left
    }

    res += (e - s);
}

return res;
```



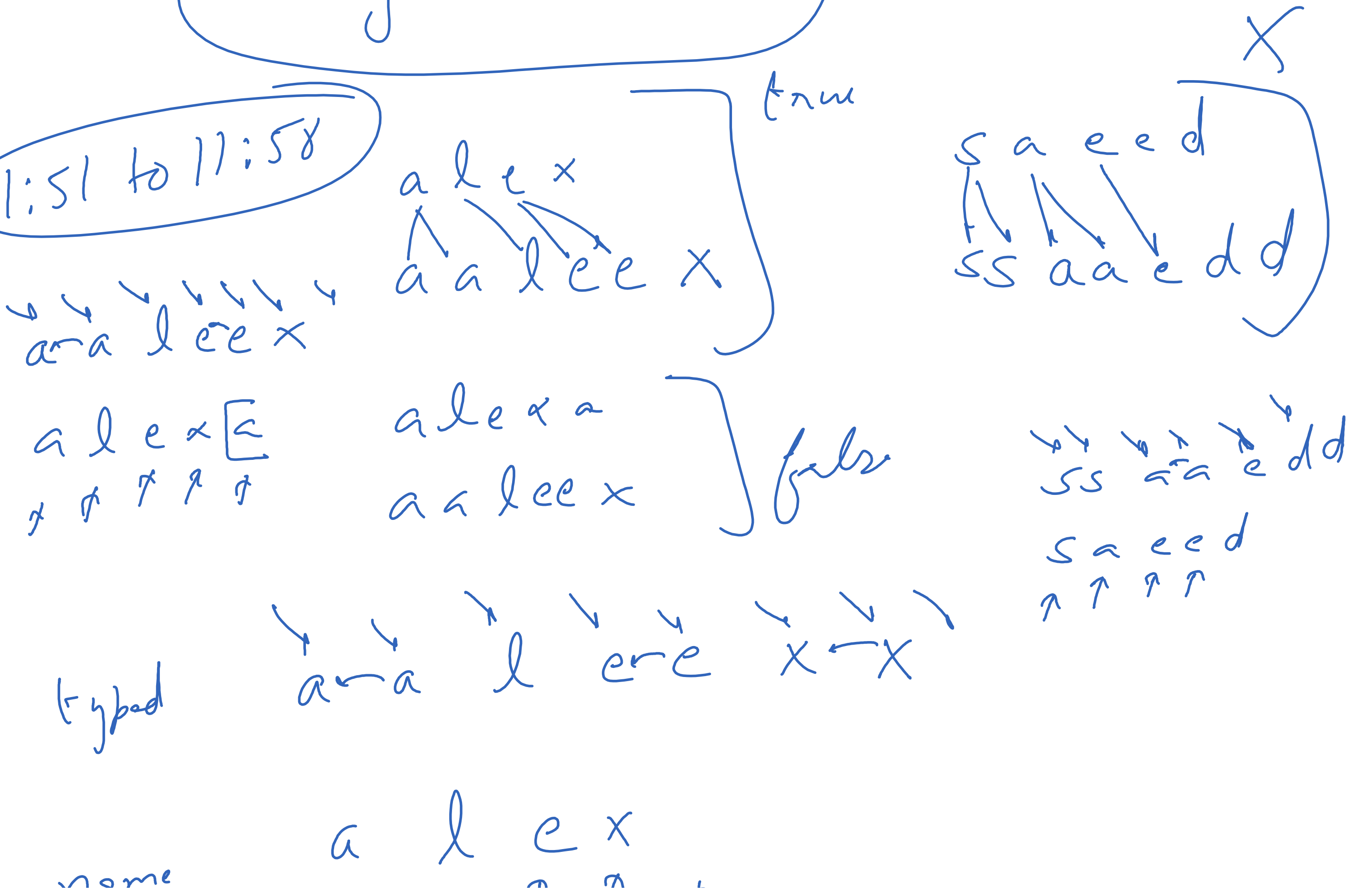


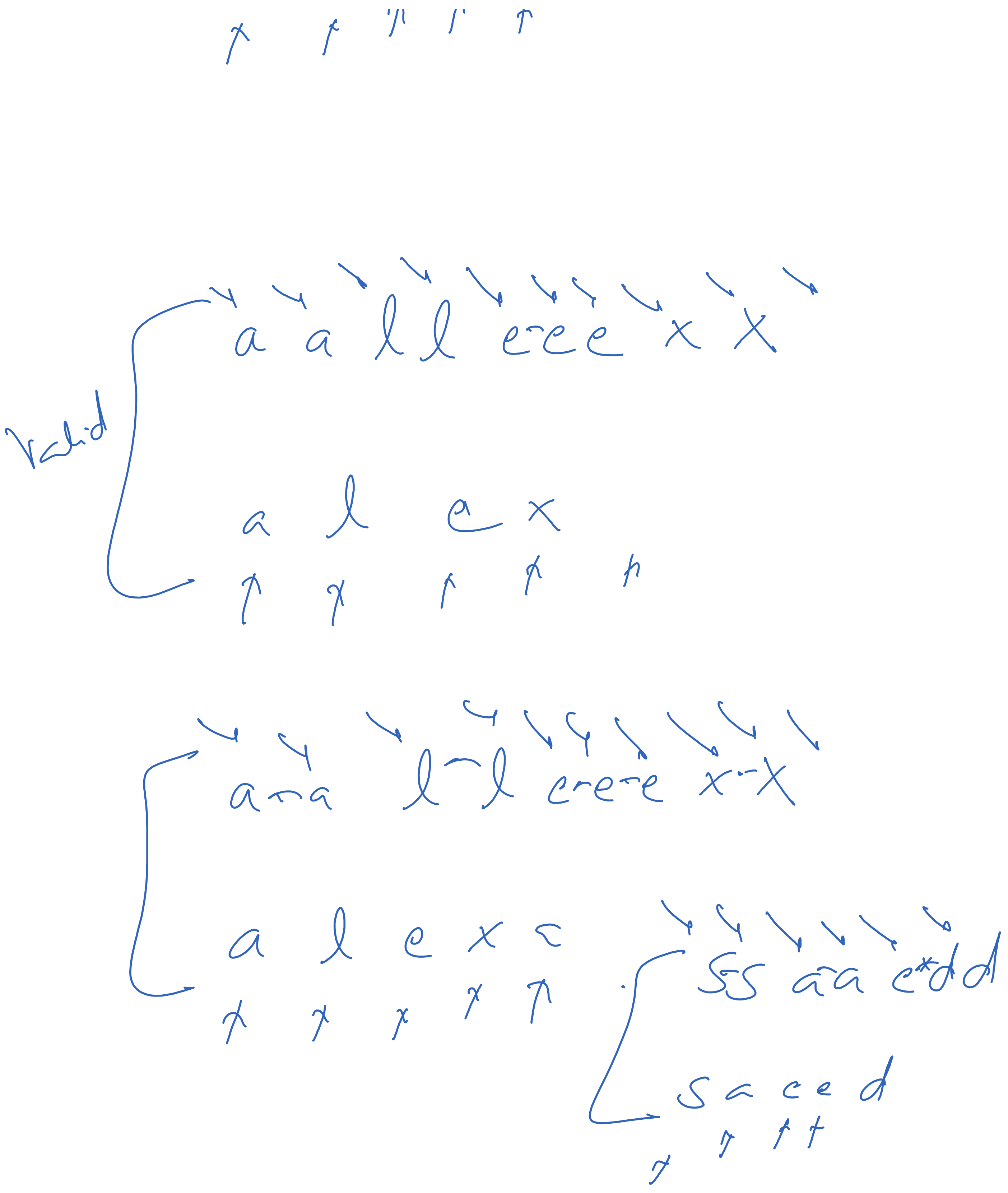
```
int s = -1;
int e = -1;

int res = 0;
for(int i = 0; i < nums.length; i++){
    if(nums[i] >= left && nums[i] <= right){
        e = i;
    } else if(nums[i] > right){
        e = s = i;
    } else {
        // lesser than left
    }

    res += (e - s);
}

return res;
```



Long Pressed

11:51 to 11:58




a a l l e e e x x
Valid
a l e x
a a l l e e e x x
a l e x c
S S a a e d d
s a c e d